面向不平衡分类的 IDP-SMOTE 重采样算法

盛　凯，刘　忠，周德超，冯成旭

(海军工程大学 电子工程学院，武汉 430033)

摘　要：传统的分类算法在对不平衡数据进行分类时，容易导致少数类被错分。为了提高少数类样本的分类准确度，提出了一种基于改进密度峰值聚类的采样算法 IDP-SMOTE。首先，采用 Box-Cox 变换和 σ 准则对密度峰值聚类算法进行改进，实现了聚类中心和离群点的自动判别；然后，将改进的密度峰值聚类算法与 SMOTE 升采样算法相结合，去除噪声数据，并基于少数类样本的局部密度和邻近距离，在子类的范围内合成采样数据。该算法有效避免了升采样导致的边界模糊，改善了类内不平衡及边界样本难以学习的问题，同时实现了自动聚类和重采样，防止了人为因素干扰。通过实验对比，验证了提出算法的有效性和自适应性。

关键词：不平衡数据；分类；重采样；密度峰值聚类

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2017.07.0699

IDP-SMOTE resampling algorithm for imbalanced classification

Sheng Kai, Liu Zhong, Zhou Dechao, Feng Chengxu

(*College of Electronic Engineering*, *Naval University of Engineering*, *Wuhan Hubei* 430033, *China*)

Abstract: When classifying imbalanced data, traditional classification algorithms are easy to misclassify the minority samples. In order to improve the classification accuracy of the minority samples, this paper proposed a novel resampling algorithm based on the improved density peaks clustering method, named IDP-SMOTE. First, improved density peaks clustering algorithm by utilizing Box-Cox transformation and σ-rule for finding the clustering centers and outliers automatically; second, combined the improved density peaks clustering algorithm with SMOTE method. With removing the noisy data, the synthetic samples can be generated in the sub-class regions on the basis of the values of local density and nearest distance of the minority samples. The presented algorithm avoids the boundary ambiguity caused by over-sampling, improves the imbalance problem with-in class and reduces the learning difficulty of the boundary data. Meanwhile, it realizes automatic clustering and resampling, and avoids the interference of subjective factors. Through the contrast experiment, the proposed algorithm is effective and adaptive.

Key Words: imbalanced data; classification; resampling; density peaks clustering

0　引言

分类是机器学习和数据挖掘领域中获取知识的重要手段之一，其目标是利用类别已知数据构建分类模型，并对类别未知数据进行预测。常见的分类算法通常假定用于训练的数据集各类平衡，即各类的样本数量大致相等。但是在获得的真实数据中，常常存在某个类别的样本数远少于其他类别的情况，此时若直接使用以最大总体分类精度为目标的分类算法进行训练，得到的分类模型往往偏向于多数类，增大了少数类被错分的可能性。然而，在异常检测、疾病诊断等实际应用中，少数类通常更加受到人们重视，其错分带来的代价也更为严重。因此，不平衡数据的分类问题成为了近年来机器学习的一个研究热点[1]。

升采样是通过增加少数类样本，达到少数类与多数类样本基本平衡，从而提高少数类分类性能的方法。简单的随机升采样通过复制少数类样本使得两类数据平衡，但这可能会导致严重的过拟合现象发生。2002 年，Chawla 等人[2]提出了 SMOTE 升采样方法。它首先通过 KNN 算法搜索少数类中每个样本的 k 个最近邻样本，然后在 与这些邻近样本之间的连线上随机取点，生成没有重复的新少数类样本集合。但是该方法没有考虑样本的分布，在邻近样本的选择上也具有一定的盲目性，导致生成新样本时容易造成正负类边界模糊以及插入噪声等问题。2008 年，He 等人[3]针对少数类样本分布不均衡问题，提出了基于样本密度分布的 ADASYN 算法，对于密度小的少数类样本合成更多的新样本数据，以减少少数类内不平衡分布导致的偏差。但是，该方法仍会导致合成的采样数据落在多数类的分布范围内。为了解决这个问题，多种基于聚类的采样技术相继被提出。

作者简介：盛凯（1991-），男，山东兰陵人，博士研究生，主要研究方向为机器学习、轨迹数据分析（shengkai0214@foxmail.com）；刘忠（1963-），男，教授，博士，主要研究方向为系统工程；周德超（1972-），男，副教授，博士，主要研究方向为智能计算、模式识别；冯成旭（1986-），男，讲师，博士，主要研究方向为轨迹数据挖掘.

例如，2011 年，Barua 等人[4]提出了基于层次聚类的 CBSO 算法；2012 年，Bunkhumpornpat 等人[5]基于 DBSCAN 算法提出了 DB-SMOTE 算法；2014 年，Cao 等人[6]提出了基于高斯混合模型的升采样算法；2015 年，Chen 等人[7]提出了改进的基于 K-means 聚类的 KM-SMOTE 算法等。这些方法旨在少数类的各个子类中进行升采样，但是计算复杂度较高，且均需要提前人工设置聚类参数，这对于处理分布未知的数据集非常困难。

针对上述问题，本文将密度峰值聚类（clustering by fast search and find of density peaks, DP）算法与传统的 SMOTE 升采样算法有机结合，提出了一种新的重采样算法 IDP-SMOTE。首先，本文对 DP 算法进行了改进，给出了聚类中心和离群点的自动判别标准；然后，根据改进的密度峰值聚类算法对各类样本聚类，并去除噪声；第三，为防止合成的少数类样本落入多数类的范围内，根据聚类结果在少数类的子类中合成少数类样本数据。考虑到类内不平衡问题以及边界样本对于分类更为重要，根据局部密度对每个少数类样本的升采样权重进行了调整，赋予了少数子类及边界附近的样本更高的升采样权重。

1　改进的密度峰值聚类算法

Rodrigues 等人[8]于 2014 年提出了 DP 聚类算法。该算法基于两条基本假设，一是聚类中心是周围邻居点中密度最大的点；二是不同聚类中心之间的距离较远。其核心步骤如下：

a)计算每个样本点 x_i 的局部密度 ρ_i。其计算公式如下：

$$\rho_i = \sum_{j\neq i}\chi(d_{ij} - d_c) \tag{1}$$

其中: d_{ij} 表示样本 x_i 与 x_j 之间的距离；d_c 为截断距离，通常可选取为所有样本间距升序排列的 1%或 2%分位数；$\chi(x)$ 为截断函数：

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \tag{2}$$

当样本点数量较少时，可采用指数核计算局部密度，其公式为

$$\rho_i = \sum_{j=1, j\neq i}^{N} \exp(-(\frac{d_{ij}}{d_c})^2) \tag{3}$$

b)计算每个样本点 x_i 与最近高密度点之间的距离（邻近距离）δ_i。对于全局密度最大的点 $x_{\max}$，其邻近距离为相对 $x_{\max}$ 的全局最大距离。因此，δ_i 的计算公式为

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j > \rho_i}(d_{ij}), & if\ \exists j\ s.t.\ \rho_j > \rho_i \\ \max\limits_{j:\rho_j > \rho_i}(d_{ij}), & otherwise. \end{cases} \tag{4}$$

c)分别以 ρ_i 和 δ_i 作为横坐标和纵坐标绘制决策图。由于聚类中心的局部密度 ρ_i 和邻近距离 δ_i 都比较大，而离群点的邻近距离 δ_i 较大，局部密度 ρ_i 非常小。因此，可通过决策图选定聚类中心和离群点。

d)指派除了聚类中心点和离群点之外的样本类别，使之与所有局部密度大于自己的点中，距离最近的样本类别相同，完成聚类过程。

DP 算法虽然不再需要指定聚类参数，但是其最终聚类中心及离群点的确定（步骤 c））却依赖于人工选择。针对此问题，文献[9]采用模糊规则实现了聚类中心的自动判别，但是样本点局部密度和邻近距离的数据分布与样本本身的数据分布相关，并不一定符合正态分布。本文在此基础上，引入统计经济学中的 Box-Cox 变换[10]，首先将局部密度 ρ_i 和邻近距离 δ_i 变换为近似正态分布；然后，为了避免密度较小的子类漏检，采用 σ 准则定义聚类中心和噪声点的判别规则。

假设有正序列 $X = \{x_0, x_1, ..., x_{N-1}\}$，Box-Cox 变换公式如下：

$$x_i = \begin{cases} \dfrac{x_i^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \ln(x_i), & \lambda = 0 \end{cases} \tag{5}$$

其中：λ 为变换参数，可采用最大化对数似然函数确定其最优值：

$$f(x,\lambda) = -\frac{N}{2}\ln\left[\sum_{i=0}^{N-1}\frac{(x_i(\lambda) - \bar{x}(\lambda))^2}{N}\right] + (\lambda - 1)\sum_{i=0}^{N-1}\ln(x_i) \tag{6}$$

其中，$\bar{x}(\lambda) = \frac{1}{N}\sum_{i=0}^{N-1}x_i(\lambda)$。

定义聚类中心和噪声点的判别规则如下：

$$EC_i = [\delta_i' \geq \mu(\delta_i') + 3\cdot\sigma(\delta_i')] \cap [\rho_i' \geq \mu(\rho_i') - \sigma(\rho_i')] \tag{7}$$

$$EN_i = [\delta_i' \geq \mu(\delta_i') + 2\cdot\sigma(\delta_i')] \cap [\rho_i' < \mu(\rho_i') - 2\cdot\sigma(\rho_i')] \tag{8}$$

其中，δ'、ρ_i' 为经 Box-Cox 变换后的 δ_i 和 ρ_i 值，EC_i 表示聚类中心，EN_i 表示噪声数据，μ 表示均值，σ 表示标准差。因此，改进的 DP 算法如算法 1 所示。

算法 1: Improved-DP

Input: 待聚类数据集 D_C

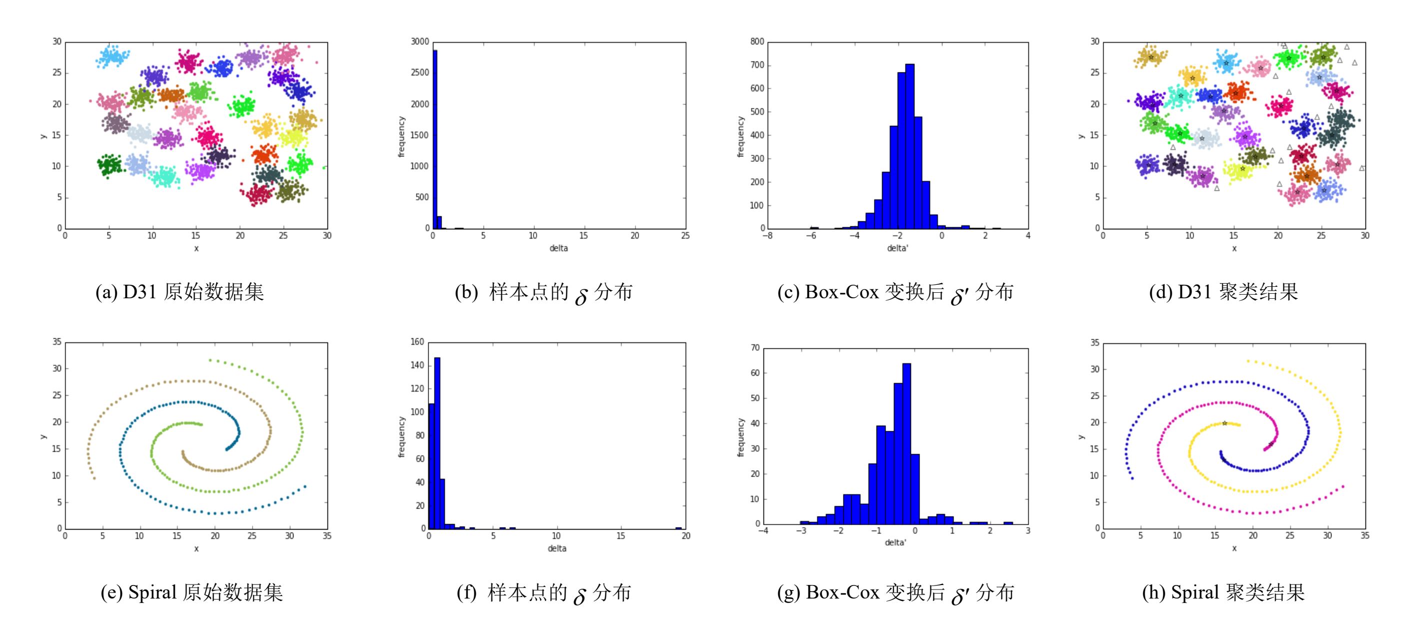
Output: 类别标号 idxC，局部密度 ρ，邻近距离 δ

procedure:

1. 计算所有样本点两两之间的距离 d_{ij}；
2. 将 d_{ij} 从小至大排序，取 d_{ij} 的 1%或 2%分位数作为截断距离 d_c；
3. 根据式(1)~(4)计算每个点的局部密度 ρ_i 及邻近距离 δ_i；
4. 根据式(5)(6)对 ρ 和 δ 进行 Box-Cox 变换，得到 ρ' 和 δ'；
5. 根据式(7) (8)确定聚类中心 EC_i 和噪声点 EN_i，标记 idxC；
6. 指派其他样本点的类别，完成聚类过程。

End

Improved-DP 算法的输入数据只有待聚类的数据集 D_C，输出数据包括每个样本点的类别标号 idxC，每个点的局部密度 ρ 及邻近距离 δ。图 1 给出了在部分数据集中 Improved-DP 算法的聚类效果，其中：(a)(e)分别是 D31 数据集和 Spiral 数据集的样本分布情况；(b)(f)分别是 D31 和 Spiral 数据集中样本点邻近距离的分布情况；(c)(g)分别是经过 Box-Cox 变换后的邻近距离分布情况，可见经过调整后，更加符合正态分布；(d)(h)展现了最终的聚类结果，图中聚类中心点和离群点分别用符号△和☆表示。可见，Improved-DP 算法聚类效果良好，且具有较强的自适应性。



(a) D31 原始数据集 (b) 样本点的 δ 分布 (c) Box-Cox 变换后 δ' 分布 (d) D31 聚类结果

(e) Spiral 原始数据集 (f) 样本点的 δ 分布 (g) Box-Cox 变换后 δ' 分布 (h) Spiral 聚类结果

图 1 Improved-DP 算法聚类效果

2 基于 Improved-DP 的重采样算法

为了能够根据数据样本分布自动确定子类，并在少数类的子类内部进行升采样，且避免类内不平衡及噪声数据的影响，本节提出了一种基于 Improved-DP 聚类算法的不平衡数据重采样算法 IDP-SMOTE，如算法 2 所示。

算法 2: IDP-SMOTE

Input: 训练集 D，升采样系数 β

Output: 重采样训练集 $D_{Resample}$

procedure:

1. 定义 D_S 与 D_L 分别为训练集 D 中少数类（正类）子集和多数类（负类）子集；

2. 对各个类进行 Improved-DP 聚类，排除噪声点，获得相应子类及各样本点的 ρ_i 和 δ_i；

3. 确定少数类的采样总数量 G。根据去除噪声点后的多数类样本数量 m_{ls} 和少数类样本数量 m_{ss}，可得：

$$G = (m_{ls} - m_{ss}) \times \beta \tag{9}$$

其中，$\beta \in [0,1]$ 为升采样系数。当 $\beta = 1.0$ 时表示升采样后训练集中的正负类样本完全平衡；

4. 确定少数类各个子类的采样数量 G_i：

$$G_i = \left(\frac{1}{m_{ssi}} \Big/ \sum_{j=1}^{n} \frac{1}{m_{ssj}} \right) \times G \tag{10}$$

其中，n 为少数类子类的个数，m_{ssi} 为第 i 个子类的样本数；

5. 计算少数类中每个样本点的采样权重 r_i：

$$r_i = \frac{1}{\rho_i} / \sum_{j=1}^{m_{ssi}} \frac{1}{\rho_j} \tag{11}$$

6. 计算少数类中每个样本的升采样数量 g_i：

$$g_i = r_i \times G \tag{12}$$

7. 少数类样本升采样：

7.1. 对于每个少数类样本 x_i，根据其邻近距离 δ_i 查找 x_i 所在子类 D_{ssi} 中的全部邻近样本；

7.2. 随机选择 x_i 的一个邻近样本 x_{is}，根据式(11)随机合成一个少数类样本 s，并重复执行 g_i 次；

$$s = x_i + \text{rand}(0,1) * (x_{is} - x_i) \tag{13}$$

8. 生成重采样后的训练集 $D_{Resample}$。

end

根据第 2 节的所述，Improved-DP 算法不但能够确定聚类簇数、聚类中心，也能发现噪声点。基于此，本文采用 Improved-DP 算法分别对多数类和少数类样本进行聚类，并排除噪声数据对分类的影响，同时，保留少数类样本聚类时计算所得的邻近距离 δ_i 及局部密度 ρ_i 等参数（第 2 步）；第 3 步和第 4 步考虑了类内不平衡对分类的影响，并借鉴文献[6]的思想，将每个子类中包含的样本数量的反比作为该子类的升采样数量；另据研究表明，边界附近的样本对于分类更为重要[3]，因此在升采样时有必要对边界附近的样本赋予更高的采样权重。由于 ρ_i 值越小，表示其样本 x_i 越可能处于边界位置，因此在算法第 5 步中，将 ρ_i 的倒数作为每个少数类样本 x_i 的采样权重，并进行归一化处理，可知，$\sum r_i = 1$；然后，于第 6 步中计算出各个少数类样本对应的升采样数量 g_i；第 7 步借鉴了 CBSO 算法[4]的升采样过程，不同的是本文在选择子类内邻近样本时采用的是距离阈值 δ_i，而不是 *k-NN* 算法中的固定个数，从而避免了 k 值选取中的人工干预。

3 实验

3.1 数据集

本文选取 UCI 机器学习数据库[11]中的 7 个数据集进行测试，分别为 Abalone、German、Glass、Leaf、Letter、Vehicle 和 Wine。由于本文只针对两类数据进行测试，需要对数据集中的多类数据进行转换。测试所用数据集的描述如表 1 所示。

表 1　数据集描述

数据集	少数类	多数类	不平衡比
Abalone	类'18'	类'9'	42:689
German	类'bad'	其他	300:700
Glass	类'5,6,7'	其他	51:163
Leaf	类'12,13,14'	其他	37:303
Letter[1]	类'A,B,C,D'	其他	3096:16904
Letter[2]	类'D'	其他	805:19195
Vehicle	类'Opel'	其他	212:634
Wine	类'1'	其他	59:119

3.2　性能评估指标

在评估不平衡数据的分类性能时，常用的评价指标包括采用 F-measure、G-means 和 AUC 等[1]。F-measure 和 G-means 的定义需要用到混淆矩阵的概念，如表 2 所示。

表 2　混淆矩阵

真实类别	预测结果	
	正类（少类）	负类（多类）
正类（少类）	TP	FN
负类（多类）	FP	TN

如果一个实例为正类并且也被预测为正类，即为真正类（TP），如果实例是负类而被预测为正类，称之为假正类（FP），类似的，其余两种情况分别为假负类（FN）和真负类（TN）。F-measure 是一种针对少数类识别性能的评价准则，其定义如下：

$$F-measure=\frac{Recall\cdot Precision}{Recall+Precision} \tag{14}$$

其中：$Recall=\frac{TP}{TP+FN}$，$Precision=\frac{TP}{TP+FP}$。F-measure 同等看待查全率 Recall 和查准率 Precision 对分类器评测的贡献。只有 Recall 和 Precision 的值均较高时，F-measure 值才能较大。

G-means 是一种衡量不平衡数据分类效果的准则，其定义如下：

$$G-means=\sqrt{TPR\cdot FPR} \tag{15}$$

其中：$TPR=\frac{TP}{TP+FN}$，$FPR=\frac{TN}{TN+FP}$。G-means 综合考虑了两类的分类准确率。如果分类器偏向于某一类，则 G-means 值将很小。

另外，AUC 也是一种非常具有区分度的评价不平衡分类器性能的方法。它表示的是 ROC（Receiver Operating Characteristic）曲线的线下面积，是代替 ROC 曲线的一种定量描述方法。AUC 的取值范围在 0 和 1 之间，取值越大说明分类器性能越好。

3.3　实验结果

本文采用随机森林分类算法对本文提出的重采样方法进行测试和评价，并与 SMOTE 算法、KM-SMOTE 算法和 DB-SMOTE 算法等升采样方法进行对比。其中，分类算法采用基于 Python 的 scikit-learn 机器学习包[12]中的标准模型和默认参数。重采样算法中，SMOTE、KM-SMOTE 以及 DB-SMOTE 的最近邻系数均设置为 k = 5；设置 KM-SMOTE 的聚类系数 c = 2；设置 DB-SMOTE 的领域半径 ε = 0.5，邻域内最少样本数 $MinPts$ = 5；所有升采样算法的升采样比例均设置为 $\beta=1$。分类器性能的采用 F-measure、G-means 和 AUC 等不平衡分类中常用的指标进行评价。每次实验随机选取 75%的各类样本作为训练集，剩余样本作为测试集。通过 100 次重复实验，获得各个指标的平均值，其结果如表 3 所示。

根据实验结果可得，SMOTE 及改进方法在大多数不平衡数据集中都比直接采用分类算法分类性能更高。然而，由于 SMOTE 算法在插值时没有考虑样本的分布；KM-SMOTE 和 DB-SMOTE 算法虽然考虑了子类分布问题，但是聚类参数的设定严重依赖于研究人员对数据集的掌握程度。当参数设置不准确时，容易造成聚类结果与实际偏差较大。这些问题都可能导致部分合成样本落在多数类的范围内，从而影响分类学习效果，甚至导致分类性能更低。本文提出的 IDP-SMOTE 采样算法在聚类和选择邻近样本时避免了主观输入参数的影响，同时去除样本噪声，并通过调整少数类样本的采样系数来改善类内不平衡和边界数据不容易被学习等问题。这些措施使得 IDP-SMOTE 采样算法在对不平衡数据分类时具有更强的适应性，在所测试的各个数据集中提升效果都比较明显，总体优胜次数最高。

4　结束语

针对不平衡数据的分类问题，本文提出了一种新的重采样算法 IDP-SMOTE。该算法能够根据数据的空间分布，更加智能的合成少数类样本，从而提高不平衡数据的分类性能。相比之前的多种采样算法，本文提出的算法同时具有以下优势：1、采用 Improved-DP 算法进行聚类，聚类簇不受空间形状限制，且避免了手动输入参数造成的主观因素干扰；2、剔除各类的噪声数据，避免噪声干扰；3、在少数类的子类内部进行升采样，避免了合成的少数类样本落入多数类的范围中；4、通过调整采样系数，改善了类内不平衡和少数类边界样本难以学习等问题，对各类不平衡数据的适应性更强。未来可对多分类问题的采样策略进行研究，使本文提出的算法能够在实际数据集中得到更广泛的应用。

表 3 基于不同采样方法的分类器性能比较

数据集	评价标准	Base	SMOTE	KM-SMOTE	DB-SMOTE	IDP-SMOTE
	F-measure	0.2624	0.3680	0.3660	0.3548	**0.3821**
Abalone	G-means	0.3706	0.5910	0.5930	0.5819	**0.6556**
	AUC	0.5814	0.6689	0.6683	0.6614	**0.7048**
	F-measure	0.4264	0.3533	0.3603	0.0513	**0.4825**
German	G-means	0.5420	0.4788	0.4841	0.1628	**0.5936**
	AUC	0.6227	0.5928	0.5968	0.5104	**0.6534**
	F-measure	0.8816	0.8840	0.8819	0.8839	**0.8915**
Glass	G-means	0.9136	0.9244	0.9285	0.9095	**0.9336**
	AUC	0.9221	0.9233	0.9260	0.9137	**0.9346**
	F-measure	0.4841	0.6370	0.6290	**0.6390**	0.6046
Leaf	G-means	0.5774	**0.7890**	0.7817	0.7835	0.7853
	AUC	0.6729	**0.8071**	0.8011	0.8033	0.8014
	F-measure	0.9368	0.9365	0.9401	0.9377	**0.9532**
Letter[1]	G-means	0.9429	0.9450	0.9462	0.9461	**0.9558**
	AUC	0.9444	0.9462	0.9475	0.9474	**0.9561**
	F-measure	0.8981	0.9220	0.9221	**0.9232**	0.9139
Letter[2]	G-means	0.9118	0.9433	0.9467	**0.9488**	0.9377
	AUC	0.9156	0.9449	**0.9482**	0.9436	0.9396
	F-measure	0.4834	0.4995	0.4720	0.4934	**0.5343**
Vehicle	G-means	0.6105	0.6466	0.6231	0.6395	**0.6855**
	AUC	0.6574	0.6667	0.6495	0.6628	**0.6938**
	F-measure	0.9752	0.9768	0.9762	0.9714	**0.9780**
Wine	G-means	0.9766	0.9735	0.9815	0.9755	**0.9839**
	AUC	0.9802	0.9815	0.9813	0.9749	**0.9843**
总体优胜次数		0	2	1	3	**18**

参考文献：

[1] 李勇，刘战东，张海军．不平衡数据的集成分类算法综述 [J]. 计算机应用研究, 2014, 31 (5): 1287-1291.

[2] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique [J]. Journal of Artificial Intelligence Research, 2002, 16 (1): 321-357.

[3] He Haibo, Bai Yang, Garcia E A, et al. ADASYN: adaptive synthetic sampling approach for imbalanced learning [C]// Proc of IEEE International Joint Conference on Neural Networks. 2008: 1322-1328.

[4] Barua S, Islam M M, Murase K. A novel synthetic minority oversampling technique for imbalanced data set learning [C]// Proc of International Conference of Neural Information Processing. 2011: 735-744.

[5] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. DBSMOTE: density-based synthetic minority over-sampling technique [J]. Applied Intelligence, 2012, 36 (3): 664-684.

[6] 陈斌，苏一丹，黄山．基于 KM-SMOTE 和随机森林的不平衡数据分类 [J]. 计算机技术与发展, 2015, 25 (9): 17-21.

[7] 曹鹏，李博，栗伟，等．基于概率分布估计的混合采样算法 [J]. 控制与决策, 2014, 9 (5): 815-820.

[8] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344 (6191): 1492-1496.

[9] Mehmood R, Bie Rongfang, Hussain D, et al. Fuzzy Clustering by Fast Search and Find of Density Peaks [C]// Proc of International Conference on Identification, Information, and Knowledge in the Internet of Things. 2016: 258-261.

[10] Bicego M, Baldo S. Properties of Box-Cox Transformation for Pattern Classification [J]. Neurocomputing, 2016, 218: 390-400.

[11] UCI machine learning repository [DB/OL]. [2017-06-10]. http://archive. ics. uci. edu/ml/datasets.

[12] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python [J]. Journal of Machine Learning Research, 2012, 12

(10): 2825-2830.